Digital Audio Effects Unit

Michael Bowers, Madison Cornett, Jessica Odish, Alain Sfeir
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
E-mails: mkbowers@oakland.edu, mcornett@oakland.edu, jessicaodish@oakland.edu, alainsfeir@oakland.edu

Abstract— **The purpose of this project is to create a digital audio effect unit that alters the incoming signal of a guitar via signal processing. The digital audio effects unit that is implemented on the NEXYS-4 Artix-7 FPGA Trainer Board utilizes an I2S serial communication protocol to receive and transmit audio data, and utilizes DSP to produce varying levels of distortion and an amplitude modulation-based "tremolo" effect on outgoing audio.**

Introduction

Analog devices such as guitar amplifiers and effects pedals have been used by guitar players to modulate and filter guitar signals to achieve specific tonal qualities and effects. Digital signal processing techniques have improved rapidly and in recent years, guitar players have been progressively incorporating digital audio effects units into their signal chains and these signals can now be used to effectively mimic the analog effects of traditional guitar equipment. Additionally, digital audio effects units are more portable and versatile than their analog counterparts. This project implements basic audio DSP on the FPGA to demonstrate the feasibility of an FPGA-centered effects system and to highlight the advantages of using a digital audio effects unit in the current musical instrument market.

Methodology

A. Preliminary Work

When brainstorming it was originally decided to utilize the Artix-7 onboard ADC in order to sample the incoming audio signal, then output the altered audio using PWM and the board's audio jack. It was determined that using Digilent's Pmod I2S expansion module would be more favorable. The module contains an ADC and DAC that handles the A/D and D/A audio conversions in and out. An input buffer for the guitar signal was not required since the Cirrus CS5343 ADC on the I2S2 module contains built-in high impedance sampling networks.

B. Using Digilent's Pmod I2S2 expansion module

An example project produced by Digikey [1] was used to handle the I2S2 communication protocol with the I2S Pmod expansion module. A transceiver block registers serial data into words with a data width of 24 bits. The audio data is channeled left or right by a word select clock. A PLL IP Block developed by Xilinx is used to generate a clock frequency of 11.29 MHz. This frequency is used as the master clock for the ADC, DAC, and I2S blocks. It is also used to set the serial clock and word select clock to frequencies of 2.82 MHz and 44.1 Khz, respectively (2.82MHz is ¼ of mclk and 44.1 Khz is 1/256 of mclk). 44.1 KHz is used as a standard sampling frequency because according to the Nyquist sampling theorem, the minimum sample rate to sufficiently reproduce a signal is double the highest possible frequency. Roughly 20KHz is the highest possible frequency audible by humans, which a 44.1KHz standard is able to reproduce. With no effects enabled, the I2S circuit serves as a passthrough for the clean guitar signal.

C. Audio Effects

I. Gain Control

Gain blocks are used to volume boost the incoming guitar signal. During the saturation stage, any 24 bit words within ½ to -½ of the amplitude are passed through, else the signal is clipped to ½ or -½ max amplitude, depending on polarity. During the gain stage, the word is bit shifted to double the amplitude. Since the ADC and DAC export and read digital audio data in two's complement, ½ and -½ thresholds were used to ensure that bit shifting would not cause the audio to invert polarity. Figure 1 shows an oscilloscope result of this issue that was caused by setting saturated audio to the max amplitude (7FFFFF was set instead of 3FFFFF on the positive side). At a threshold of 7FFFFF, the word would become FFFFFF after bit shifting, which is -1 in 2C. A threshold of 3FFFFF solves this issue by ensuring that the gained

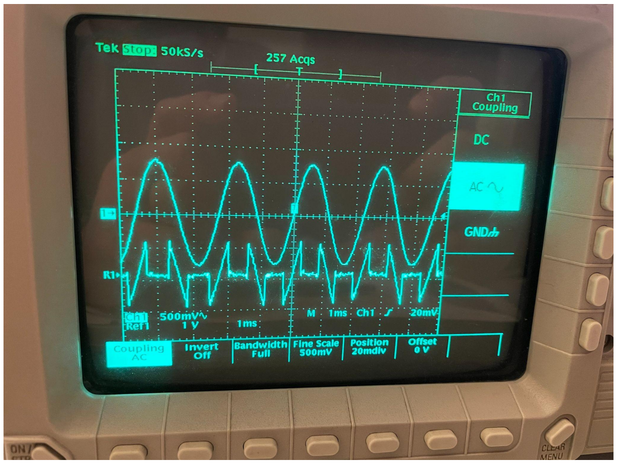

Figure 1: Wave inversion issue when clipping

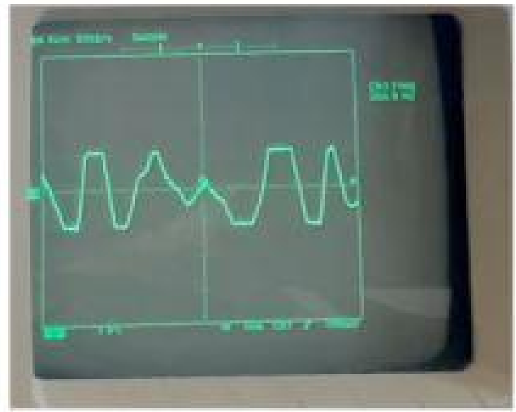

Figure 3: Hard clipping of guitar signal

value after bit shifting is 7FFFFF. The output of each gain block is muxed to allow the user to toggle the gain on or to pass the audio through using a switch. Figure 2 shows the switch scheme used to toggle the effects. Four gain blocks are connected in series to give the user a degree of volume control. While intended for volume control, the gain blocks are also sufficient for producing a clipped/distorted guitar tone for guitars with higher outputs (guitars with higher output pickups or built in active preamps).

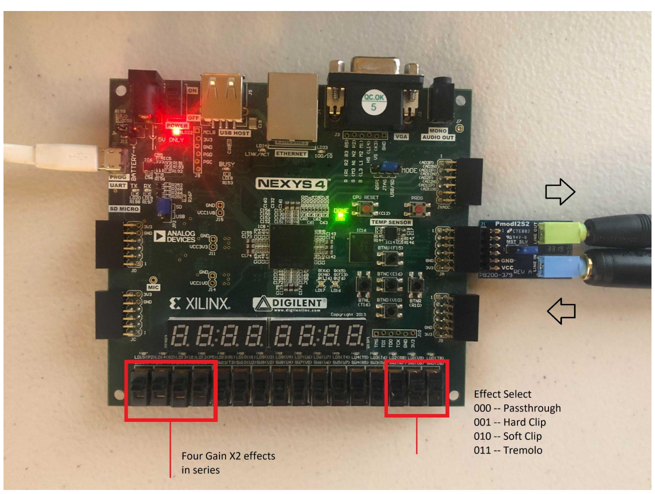

Figure 2: Control Scheme for Effects

II. Hard Clipping

Hard clipping is a common type of distortion utilized in many genres of rock music. It has a characteristic “fuzzy” or “gritty” tone due to the higher harmonics that are added to produce the hard clipping of the audio, similar to how a square wave is produced by the addition of multiple harmonic frequencies to a sine wave. Figure (3) shows an oscilloscope result of the guitar signal running through the hard clipping block. A clipping threshold of 3FFFFF was used, which was also inverted for the negative portion of the wave. Any values beyond the clipping threshold are set to the threshold, while any values under the threshold are passed through.

III. Soft Clipping

Soft clipping is associated with overdrive effects that produce gentler levels of distortion than hard clipping. Fewer higher harmonic frequencies are produced, and considered “warmer” or “smoother” than hard clipping distortion. Digital soft clipping is usually implemented with non-linear functions to produce boosted guitar signals that are more rounded than hard clipped signals.

$$f(x) = \begin{cases} 2x & \text{for } 0 \le x < 1/3 \\ \frac{3-(2-3x)^2}{3} & \text{for } 1/3 \le x < 2/3 \\ 1 & \text{for } 2/3 \le x \le 1 \end{cases}$$

Figure 4: Soft Clipping Equation

Figure 4 shows the three layer soft clipping scheme [2] that was used to implement the effect. A boost effect was used for values from 0 to ⅓ amplitude. A nonlinear function was used for values from ⅓ to ⅔ amplitude, and a hard clipper was set for values above ⅔ amplitude. A LUT was used to store soft clipping results for values ranging from -128 to +127 (80 to 7F). For ease of implementation, the top 8 bits of the 24 bit word were considered while the rest were passed through. The top 8 bits of the input word served as the address for the memory block position of the distortion result value associated with the value of the address. All values in the LUT were calculated using Microsoft Excel. [3] was referenced to utilize block ram to store the LUT values.

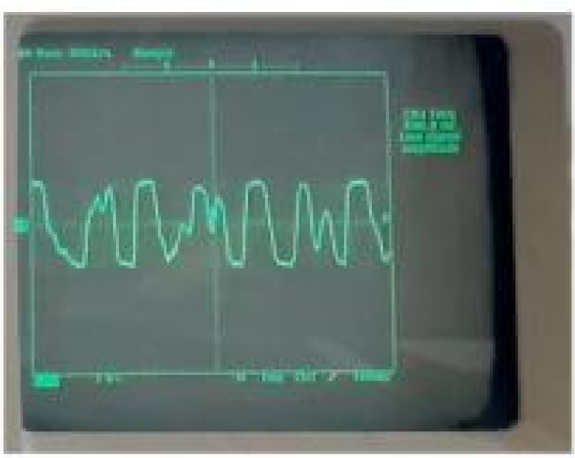

Figure 5: Soft Clipping of Guitar Signal

Figure 5 shows an oscilloscope result of the guitar signal running through the soft clipping block. Compared to hard clipping, soft clipping exhibits a much smoother saturation curve as the signal's amplitude increases.

IV. Tremolo

Tremolo is an amplitude modulation effect that uses a low frequency oscillator (LFO) to rapidly change the volume of the guitar signal, which sounds like a regular periodic pulsing of the guitar signal. A LFO was generated using a 512-sized LUT storing 8 bit values of a full sine wave period. The frequency of the sine wave was set by two counters. A 0-4666 counter is used to periodically assert an enable signal for a 0-511 counter. The value of the 0-511 counter corresponds to an address in the LUT, or a particular location on the sine wave. As both counters run and periodically reset, a sine wave is continually produced from the LUT. The bottom 16 bits are filled with 0s or 1s, depending if the value of the output word is positive or negative. The frequency of the sine wave is calculated as:

Freq = 11.29MHz / 4666 / 512 = 4.726 Hz

The top 8 bits of the 24 bit word of the sine wave is multiplied with the guitar signal, and the top 24 bits of the 32 bit multiplication result are taken as the resultant word. Figure 6 shows an oscilloscope result of the guitar signal running through the tremolo block.

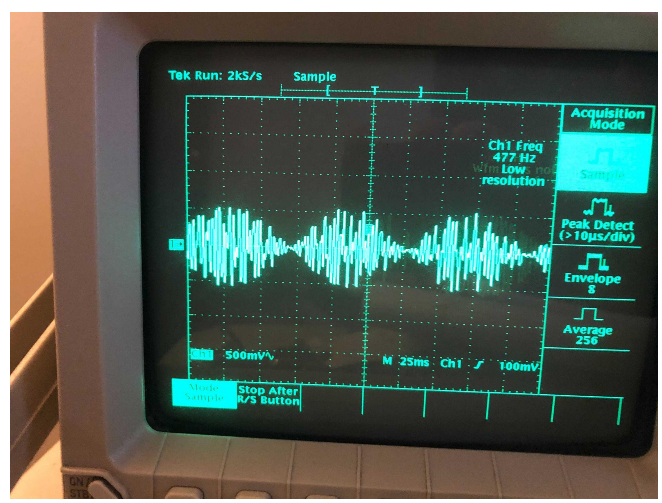

Figure 6: Amplitude Modulation effect from tremolo on a guitar signal

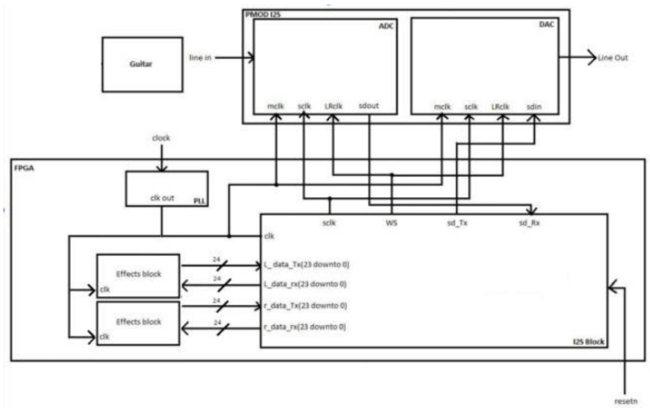

Figure 7: Top level Block diagram

V. Effect and Top Level Scheme

Figure 7 shows a block diagram of the effect chain. The guitar signal is routed through a bypassable gain series. A mux allows for switching between hard clipping, soft clipping, tremolo, and an effect bypass. Figure 8 shows the block diagram of the top file (i2s_top.vhd) + the I2S2 Pmod device. Since words are separated by left and right channels, an effects block is applied to each channel before they loop back into the I2S2 transceiver and are sent out to the DAC as serial data.

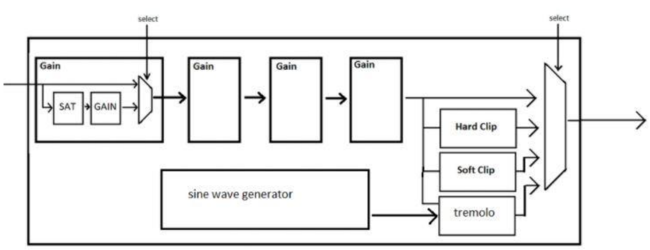

Figure 8: Effects Block Diagram

Experimental Setup

Vivado 2017.2 was used to develop and test the digital effects. An oscilloscope attached to one end of an audio splitter was used to measure the output of the circuit as it was concurrently outputted to a speaker and monitored by ear. Guitar effects were largely adjusted by ear and by regenerating the project bitstream to hear the adjusted effects and results.

A number of non-linear soft clipping functions were considered but the MATLAB code of the 3 layer soft-clipping function example was decided on as it was brought into Octave and applied to test recordings of guitar playing recorded in a digital audio workstation.

Results

The implementation of the project was successful and fully within expectations. The gain successfully boosted the incoming guitar signal and could serve as an additional source of distortion when fully toggled. The hard clipper was successful in emulating popular effects found in traditional fuzz and distortion pedals, while the soft clipper was able to mimic traditional overdrive effects. The implementation of the tremolo effect was also successful, and was beyond the original scope of the project as it was a modulation-based non-distortion effect. The tones in the effects unit are also partly customizable as there are numerous combinations of gain + effect to change the user's tone.

Challenges, Limitations and Future Work

An 8 bit width was used for the LUT of the soft clipper, which eased the calculation process of the distortion curve, but resulted in some high frequency noise coming through in the output. As the circuit snaps to the nearest quantization value as defined by the LUT (256 possible values), the wave will step up or down, and the vertical slopes of the steps represented as high frequency transients. Expanding the resolution of the LUT to a higher resolution such as 12 bits (corresponding to 4096 possible values) would minimize transient noise and produce a more accurate response curve and a more gradual softening of the soft clip. A 16-bit industry standard, or 65536 possible values, is also feasible.

A pipelined architecture for the non-linear wave component was also considered. A LUT-based solution was used instead as a simpler approach to non-linear processing. Additionally there is plenty of room to improve upon the resolution of the LUT as only two 18-Kb block RAM blocks out of a total of 270 available with the XC7A100T were being utilized.

One avoidable challenge that continually hampered the development of the guitar effects was the lack of the use of testbenches to test each audio effect. A testbench with test words (perhaps test words constituting a sine wave) could've been used to prototype the effects and see resultant words more quickly than rebuilding the project and listening to the changes, although listening and viewing the effect with the oscilloscope allowed for greater scrutiny of the guitar effect since the results of these effects are very predictable.

Additionally the effects that were chosen were relatively easy to implement compared to other effects that were considered, such as time-varying effects (reverb, chorus, phaser, etc.) or filters (LPF, bandpasses) based around finite impulse responses, which would have required much more time to implement, but could be added into the effects block in the future.

Future variants of the tremolo effect could have easily been expanded by including multiple selectable LFOs of varying shapes including sine waves, triangle waves and square waves at varying frequencies set by the user. A provision was made in the code for an adjustable frequency-setting counter for the sine wave generator but was not fully implemented. A future project revision could also include running the effects in series to give the user more options for different effect combinations, such as using the tremolo effect for post-processing of a soft clipped guitar signal.

Conclusions

This project has strengthened the team's knowledge of basic digital signal processing in VHDL and of the I2S protocol for audio devices. This project also aims to demonstrate digital effects modeling as a viable alternative to traditional analog equipment. Future work includes improving existing effects by increasing the bit depth of the LUTs and by increasing the amount of options available to the user by adding different distortion curves and additional waveform generators for the tremolo effect. Effect types that could be added include time-varying effects such as phaser and reverb and filters based around finite impulse responses.

References

[1] Digikey "I2S Pmod Quick Start (VHDL)" Digikey. [Online]. Available: https://forum.digikey.com/t/i2s-pmod-quick-start-vhdl/13065

[2] D. Marshall, "Digital Audio Effects," MATLAB DSP GRAPHICS, 2011. [Online]. Available: http://users.cs.cf.ac.uk/Dave.Marshall/CM0268/PDF/10_CM0268_Audio_FX.pdf.

[3] Xilinx, "Vivado Design Suite User Guide" 2016, [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2016_4/ug901-vivado-synthesis.pdf

[4] "I²S," Wikipedia, 31-Jan-2021. [Online]. Available: https://en.wikipedia.org/wiki/I%C2%B2S.

[5] "Pmod I2S2 Reference Manual," Pmod I2S2 Reference Manual - Digilent Reference. [Online]. Available: https://reference.digilentinc.com/reference/pmod/pmodi2s2/reference-manual.